10/577677
Express Mail No.: EV 839874032 US

IAP20 Rec'd PCT/PTO 28 APR 2006

SIGNAL TRANSITION FEATURE BASED CODING FOR SERIAL LINK

FIELD

[0001] Embodiments of the invention relate to data processing systems; and more specifically, to signal transition feature based coding for serial link.

BACKGROUND

[0002] For circuits optimized for low-power, the power dissipated at the I/O (input/output circuit) is typically around 50% of the total power consumption. This I/O power dissipation is a consequence of relatively large dimensions of devices in the I/O pads and of the external capacitances due to I/O pins, wires, and connected circuits. The devices in the I/O need to be large in order to drive the large external capacitances and this further increases their own parasitic capacitances. The capacitances on printed circuit boards are about two orders of magnitude larger than that inside of a chip. Dynamic charging and discharging these capacitances causes I/O pins to consume a relatively large amount of power.

[0003] Coding the I/O for lower power consumption has been utilized recently. Coding I/O for low power is practical and has been implemented in commercial chips. However, most of the conventional approaches are designed for parallel buses.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Embodiments of the invention may best be understood by referring to the following description and accompanying drawings that are used to illustrate embodiments of the invention. In the drawings:

[0005] Figure 1 is an audio link representing a serial link which may be used with an embodiment.

[0006] Figures 2A-2C are block diagrams illustrating different bit configurations of a data stream according to certain embodiments.

[0007] Figure 3 is a block diagram illustrating a parallel-to-serial converter.

[0008] Figure 4 is a block diagram illustrating an example of an encoder according to one embodiment.

[0009] Figure 5 is a block diagram illustrating an example of an encoder according to another embodiment.

[0010] Figure 6 is a flow diagram illustrating an example of a process for encoding data stream according to one embodiment.

[0011] Figure 7 is a block diagram illustrating an example of a device for encoding data stream according to one embodiment.

[0012] Figure 8 is a block diagram of a data processing system which may be used with an embodiment.

DETAILED DESCRIPTION

[0013] Signal transition feature based coding for serial link is described herein. In the following description, numerous details are set forth to provide a more thorough explanation for embodiments of the present invention. It will be apparent, however, to one skilled in the art, that embodiments of the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring embodiments of the present invention.

[0014] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification do not necessarily all refer to the same embodiment.

[0015] Accordingly, in one embodiment, a low power coding technique for serial links is utilized. An example of a serial link is an audio link popularly used in a variety of data processing systems. By using this coding technique, in certain configurations, approximately 12.5% of the link power consumption may be reduced only at the expense of two simple circuits added inside the chip, whose power consumption may be neglected.

[0016] Throughout this application, an audio link is utilized as an example of a serial link. However, this technique is not limited to audio links. This technique may be applied to other serial links suitable to applying this coding technique to decrease the link power consumption.

[0017] In general, the power dissipated in a CMOS (complementary metal oxide semiconductor) circuit can be classified as static power dissipation (e.g., overlap current and DC or direct current static) and dynamic power dissipation. The dynamic power dissipated by a CMOS circuit may be determined as follows:

$$P_{chip} \propto \sum_{i=1}^{N} C_{load\,i} \bullet V_{dd}^2 \bullet f \bullet p_{t_i} \quad (1)$$

[0018] Where the sum is calculated over all the N nodes of the circuits; C_{load_i} is the load capacitance at the node i; V_{dd} is the power supply voltage; f is the signal transition frequency; and p_{t_i} is the activity factor at node i. For achieving low-power in circuits one or more of the terms V_{dd}, C_{load_i} f, and p_{t_i} may be reduced. It is effective to reduce p_{t_i} in terms of power reduction when the V_{dd} and C_{load_i} are fixed. For a communication link or channel during a specific time interval, an

activity factor represents the percentage of time that a signal is present in the channel or link in either direction.

[0019] For a low power VLSI (very large-scale integration) design, the focus is on developing low power circuits without affecting too much the performance (area, latency and period). The idea behind embodiments of techniques for decreasing the power dissipation is to code the data according to the signals transition feature, in order to decrease the node's activity factor p_{t_i} on the large capacitance side (I/O) even at the expense of slightly increasing the number of transitions on the low capacitance side (e.g., internal circuits).

[0020] For example, as shown in Figure 1, the link architecture of high definition (HD) audio is the next generation audio link on data processing systems. HD audio is the serial link between chips (e.g., chipset and audio codec), which has relatively large I/O capacitances resulting in relatively large I/O power dissipation during signal transition due to dynamic capacitances charging and discharging. There are dedicated input and output serial data signals (e.g., SDIN for serial data input and SDOUT for serial data output). Each audio sample (e.g., 8bits, 16bits or 24bits per sample) goes through a parallel-to-serial conversion shifter to be transmitted over a serial link and a serial to parallel conversion shifter at the receiver side.

[0021] For audio samples with 8 bit resolution (e.g., range from 0~255), for example, more often, the samples' values are less than half of the peak value 255 (e.g., only unsigned coding are being considered). Similarly, for audio conversation, almost half of the time is quiet phase and the sample's values are even small. This means that the percentage of MSB (most significant bit) of each audio sample being 0 is relatively high (at least about 75%).

[0022] Taking the advantages of shifters for conversion between parallel and serial, as well as the high percentage of MSB of each audio sample being continued 0, according to certain embodiments, the bit's transmission sequence in each sample interlacing as shown in Figure 2A may reduce the I/O signal activity factor p_{t_i}. For

normal transmission as shown in Figure 2B, the data is transmitted by original sequence.

[0023] According to certain embodiments of the invention, certain bits of certain segments of a data stream may be coded according to certain orders. The coding orders may be determined based on the activity factors of the segments, where the activity factors may be determined at real-time when the data stream is received. In one embodiment, certain bits of certain segments of the data stream may be coded such that a number of consecutive bits having identical logic values (e.g., logical value 0 or 1) are greater than those of the sequence prior to coding. As a result, a number of transitions (e.g., activity factors) of the data stream may be reduced.

[0024] In a particular embodiment, the MSB and LSB's sequence may be converted for a first sample, then normal sequence for a second sample, and then converted, then normal, going on (e.g., every other samples converted) as shown in Figure 2C. In the converted sequence, according to one embodiment, the MSB of the first sample is coded neighboring to the MSB of the next sample. If their values are same at this time (which is 75% of the case), the signal transitions can be reduced.

[0025] For the normal original sequence, as shown in Figure 2B, MSB is neighboring to LSB. MSB has the probability of 75% being 0 and LSB's value is randomly distributed between 0 and 1. In a particular embodiment, the activity factor (also referred to as a transition factor) between MSB and LSB can be calculated as shown in Table 1 as follows.

Distribution of MSB	Distribution of LSB	Probability of no signal transition	Probability of signal transition
1 (25%) 0 (75%)	1 (50%) 0 (50%)	$\frac{1}{4}\times\frac{1}{2}+\frac{3}{4}\times\frac{1}{2}=\frac{1}{2}$	$\frac{1}{4}\times\frac{1}{2}+\frac{3}{4}\times\frac{1}{2}=\frac{1}{2}$

Table 1

[0026] For the converted sequence, MSB is neighboring to next sample's MSB once of two borders between samples. MSB has the probability of 75% being 0 and LSB's value is randomly distributed between 0 and 1. The activity factors between two MSBs and two LSBs can be calculated as shown in Table 2 as follows.

Distribution of MSB	Distribution of MSB	Probability of no signal transition	Probability of signal transition
1 (25%) 0 (75%)	1 (25%) 0 (75%)	$\frac{1}{4}\times\frac{1}{4}+\frac{3}{4}\times\frac{3}{4}=\frac{5}{8}$	$\frac{1}{4}\times\frac{3}{4}+\frac{3}{4}\times\frac{1}{4}=\frac{3}{8}$
Distribution of LSB	Distribution of LSB	Probability of no signal transition	Probability of signal transition
1 (50%) 0 (50%)	1 (50%) 0 (50%)	$\frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{2}=\frac{1}{2}$	$\frac{1}{2}\times\frac{1}{2}+\frac{1}{2}\times\frac{1}{2}=\frac{1}{2}$

Table 2

[0027] As shown above, due to LSB's value being randomly distributed, the converted sequence has relatively less impact on LSB's activity factor. However, the converted sequence may reduce MSB's activity factor approximately from $\frac{1}{2}$ to $\frac{3}{8}$ in every two sample neighbors. Thus about 12.5% signal activity factors between each sample may be reduced, according to certain embodiments.

[0028] For example, in an audio link configuration, which is typically a serial link, a data stream may be received from a parallel bus and needs to be converted into a serial data stream prior transmitting onto the serial link. Typically, a parallel-to-serial converter may be utilized to convert the data stream in parallel to a serial data stream.

[0029] Figure 3 is a parallel-to-serial converter used in a conventional audio link interface circuit. Referring to Figure 3, device 300 includes a latch circuit 301 and a shifter circuit 302. When the input data stream 304 is received in parallel, once the load signal 303 is asserted, the latch circuit 301 latches the input data stream 304. Thereafter, for every clock cycle of the clock signal 305 received, the shifter circuit

302 converts, via shifting operations, the latched parallel data stream into a serial data stream 306 to be transmitted onto a serial link. Normally, a data segment is shifted from LSB to MSB in this arrangement.

[0030] According to one embodiment, direction control logic is utilized in a parallel-to-serial converter that shifts a data segment from LSB to MSB or vice versa. Figure 4 is a block diagram illustrating an example of a parallel-to-serial converter according to one embodiment. Note that device example 400 may be implemented in hardware, software, or a combination of both.

[0031] Referring to Figure 4, similar to device 300 of Figure 3, device 400 includes a latch circuit 401 and a shifter circuit 402. When the input data stream 404 is received in parallel, the latch circuit 401 latches the input data stream 404. Thereafter, for every clock cycle, the shifter circuit 402 converts, via shifting operations, the latched parallel data stream into a serial data stream 406 to be transmitted onto a serial link.

[0032] In addition, device 400 further includes direction control logic 407 coupled to the latch circuit 401 to control a direction of bits of the latched data to be shifted by shifter module 402, in response to a direction control signal 408. The direction control signal 408 may be generated based on one or more activity factors of the data stream 404, which may be determined at real time. As a result, a data stream may be shifted from LSB to MSB or vice versa.

[0033] In one embodiment, since for serial bus, there is already a shifter (e.g., single direction, left or right) for audio sample parallel to serial conversion, a left and right direction controlled logic may be added to the shifter. To identify which samples transmission sequence has been converted, for example, after reset, the first sample is non-converted; the second is converted, and then follow this pattern.

[0034] Figure 5 is a block diagram illustrating an example of a parallel-to-serial converter according to another embodiment. For example, device 500 may be implemented as a part of device 400 of Figure 4. In this example, a data stream having an 8-bit data width is used for purposes of illustration. Referring to Figure 5, device 500 includes, but is not limited to, a latch circuit 501 and a shifter circuit 502. When the input data stream 504 is received in parallel, once the load signal

503 is asserted, the latch circuit 501 latches the input data stream 504. Thereafter, for every clock cycle of the clock signal 505 received, the shifter circuit 502 converts, via shifting operations, the latched parallel data stream into a serial data stream 506 to be transmitted onto a serial link. In one embodiment, the shifter circuit 502 may shift the latched data from left to right or vice versa, dependent upon an output of direction control logic 507 in response to a direction control signal 508. The direction control signal 508 may be determined based on one or more activity factors of the latched data. Other components may also be included.

[0035] Due to the shifter and added direction controlling logics are inside the chip, it is estimated the additional core logic's power consumption. In certain embodiments, the direction controlling core logic for the shifter may consume approximately 2.57μW of power. For equivalent capability, FPGA will consume about 20 times of power as that of ASIC. Thus the power consumption of additional logics inside the chips is about 0.1285μW.

[0036] For audio samples transmitted in normal original sequence, the signal transition percentage of MSB and LSB may be about 50%. The I/O pin capacitance of SDIN is about 7.5pF, with the clock speed of 24MHz, the V_{dd} being 3.3V, the p_{t_i} being 50%, the power consumption of the two I/O pins signal transition under normal original sequence may be determined as follows.

$$P_{chip} \propto \sum_{i=1}^{N} C_{load\,i} \bullet V_{dd}^2 \bullet f \bullet p_{t_i} \propto \sum_{i=1}^{2} 7.5pF \bullet 3.3^2 \bullet \left(24MHz \times \frac{1}{8}\right) \bullet 0.5 = 245\mu W$$

[0037] With the converted coding transmission sequence, approximately 12.5% of the MSB and LSB's transition may be reduced, thus $245\mu W \times 12.5\% = 30.6\mu W$ I/O power consumption may be reduced only at the expense of inner additional control logics of approximately 0.1285 μW power consumption, which may be neglected compared with the power consumption reduced.

[0038] Figure 6 is a flow diagram illustrating an example of a process for encoding data according to one embodiment. Process 600 may be performed by a

processing logic that may include hardware (circuitry, dedicated logic, etc.), software (such as is run on a general purpose computer system or a dedicated machine), or a combination of both. For example, process example 600 may be performed by devices 400 of Figure 4 and/or 500 of Figures 5.

[0039] In one embodiment, process 600 includes, but is not limited to, in response to a data stream to be transmitted onto a serial communication link, encoding one or more bits of the data stream according to a bit order determined based on a frequency of signal transitions of the data stream, forming a sequence in an encoded data stream having a lower number of bit transitions with respect to the frequency of signal transitions of the data stream prior to the encoding via reordering of the bits in the data stream, and transmitting the encoded data sequence onto the serial communication link.

[0040] Referring to Figure 6, processing logic receives a data stream to be transmitted over serial link, where the data stream includes a first segment and a second segment (block 602). The data stream may be part of an audio data stream, for example, received over a parallel bus. The data stream may be in a variety of format or data width according to certain embodiments, such as, for example, 8-bit, 16-bit, or 32-bit data width, etc. In response to the data stream, processing logic determines one or more activity factors of the data stream, for example, with respect to the first and second segments (block 604). An activity factor may represent bit patterns of the first and second segments (e.g., transitions from logical value of "0" to "1", or vice versa). An activity factor may be determined at real time. Alternatively, an activity factor may be embedded within the data stream, such as, for example, metadata associated with the data stream.

[0041] Processing logic rearranges the bit order of at least one of the first and second segments according to a manner determined based on the activity factor or factors (block 606). In one embodiment, the bit order of at least one of the first and second segments may be performed using one or more techniques described above. For example, a first or a second sample may be inverted prior to serializing them, such that a number of consecutive bits having identical logical values may be increased to reduce an activity factor (e.g., a number of transitions between logical

values of "0" and "1", or vice versa). Thereafter, the arranged bit order of the first and second segments are encoded forming a sequence of data steam suitable to be transmitted onto a serial link (block 608). Other operations may also be included.

[0042] Figure 7 is a block diagram illustrating an example of system configuration which may be used one or more techniques described above according to one embodiment. For example, system 700 may be implemented as a part of a data processing system having device 400 and/or device 500 of Figure 4 and/or Figure 5. System 700 may perform one or more of the operations of process 600 of Figure 6.

[0043] Referring to Figure 7, system 700 includes, but is not limited to, a chipset 701 having a memory controller hub (MCH) 706 and an IO (input/output) controller hub (ICH) 707. MCH 706 may be coupled to a main memory, which may be a DRAM (dynamic random-access memory), or the like. ICH 707 may include serial communication logic 708, which may include some or all of device 400 if Figure 4 and/or device 500 of Figure 5. In one embodiment, the serial communication logic 708 may be coupled to one or more serial communication interface devices 704-705 via a serial communication link 703, which may be a bus or an interconnect dependent upon communication protocols, etc. In on embodiment, the serial communication logic may perform some or all of the operations of process 600 of Figure 6.

[0044] In one embodiment, chipset 701 includes, but is not limited to, an input/output (I/O) circuit having an associated encoder to encode one or more bits of the data stream according to a bit order determined based on a frequency of signal transitions of a data stream, in response to the data stream to be transmitted onto a serial communication link. The encoder is configured to form a sequence of encoded data stream having a lower number of bit transitions with respect to the frequency of signal transitions of the data stream prior to the encoding via reordering of the bits in the data stream. The device further includes a serial communication interface coupled to the encoder to transmit the encoded data sequence onto the serial communication link. Other configurations may exist.

[0045] Figure 8 is a block diagram of an example computer system that may use an embodiment having at least one of the features described above. In one embodiment, computer system 800 includes a communication mechanism, bus, or interconnect 811 for communicating information, and an integrated circuit component such as a main processing unit 812 coupled with bus 811 for processing information. One or more of the components or devices in the computer system 800 such as the main processing unit 812 or a chip set 836 may use an embodiment of the techniques described above. The main processing unit 812 may consist of one or more processors or processor cores working together as a unit.

[0046] Computer system 800 further includes a random access memory (RAM) or other dynamic storage device 804 (also referred to as a main memory) coupled to bus 811 for storing information and instructions to be executed by main processing unit 812. Main memory 804 may also be used for storing temporary variables or other intermediate information during execution of instructions by main processing unit 812.

[0047] Firmware 803 may be a combination of software and hardware, such as Electronically Programmable Read-Only Memory (EPROM) that has the operations for the routine recorded on the EPROM. The firmware 803 may embed foundation code, basic input/output system code (BIOS), or other similar code. The firmware 803 may make it possible for the computer system 800 to boot itself.

[0048] Computer system 800 may also include a read-only memory (ROM) and/or other static storage device 806 coupled to bus or interconnect 811 for storing static information and instructions for main processing unit 812. The static storage device 806 may store OS level and application level software. Computer system 800 may further be coupled to a display device 821, such as a cathode ray tube (CRT) or liquid crystal display (LCD), coupled to bus 811 for displaying information to a computer user. A chipset may interface with the display device 821.

[0049] An alphanumeric input device (keyboard) 822, including alphanumeric and other keys, may also be coupled to bus 811 for communicating information and command selections to main processing unit 812. An additional user input device is

cursor control device 823, such as a mouse, trackball, trackpad, stylus, or cursor direction keys, coupled to bus 811 for communicating direction information and command selections to main processing unit 812, and for controlling cursor movement on a display device 821. A chipset may interface with the input output devices.

[0050] Another device that may be coupled to bus 811 is a hard copy device 824, which may be used for printing instructions, data, or other information on a medium such as paper, film, or similar types of media. Furthermore, a sound recording and playback device, such as a speaker and/or microphone (not shown) may optionally be coupled to bus 811 for audio interfacing with computer system 800. Another device that may be coupled to bus 811 is a wired/wireless communication capability 825.

[0051] According to one embodiment, chipset 836 may include some or all of the device 400 of Figure 4 and/or device 500 of Figure 5, which may performed one or more operations involved in process 600 of Figure 6. Other components may also be included.

[0052] Thus, signal transition feature based coding for serial link has been described herein. Some portions of the preceding detailed descriptions have been presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the ways used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0053] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely

convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the above discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0054] Embodiments of the present invention also relate to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), erasable programmable ROMs (EPROMs), electrically erasable programmable ROMs (EEPROMs), magnetic or optical cards, or any type of media suitable for storing electronic instructions, and each coupled to a computer system bus.

[0055] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method operations. The required structure for a variety of these systems will appear from the description below. In addition, embodiments of the present invention are not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of embodiments of the invention as described herein.

[0056] A machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For

example, a machine-readable medium includes read only memory ("ROM"); random access memory ("RAM"); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0057] In the foregoing specification, embodiments of the invention have been described with reference to specific exemplary embodiments thereof. It will be evident that various modifications may be made thereto without departing from the broader spirit and scope of embodiments of the invention as set forth in the following claims. The specification and drawings are, accordingly, to be regarded in an illustrative sense rather than a restrictive sense.